CMSC 28100

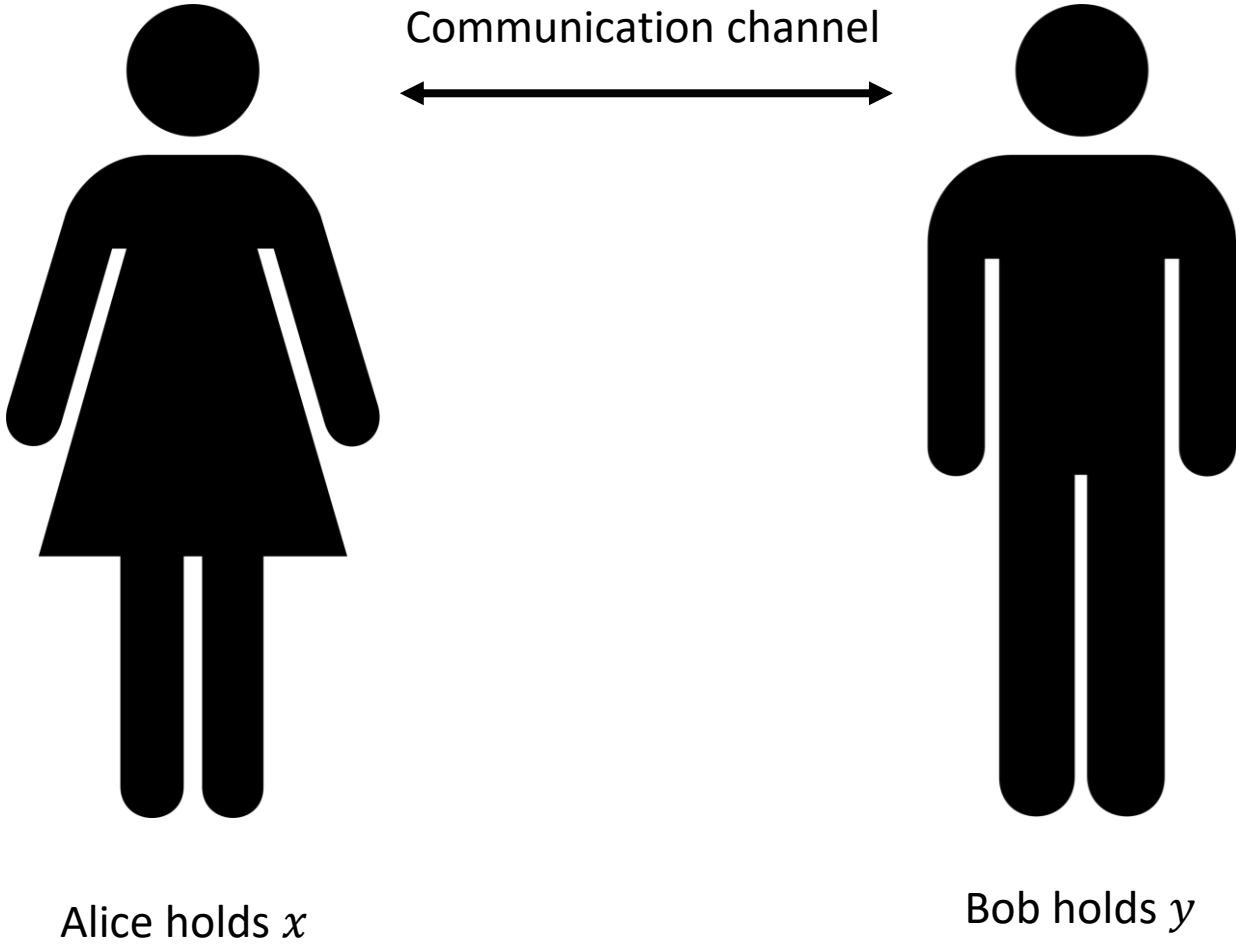
Introduction to Complexity Theory

Spring 2025
Instructor: William Hoza

Communication Complexity

Communication complexity

- Goal: Compute $f(x, y)$ using as little communication as possible



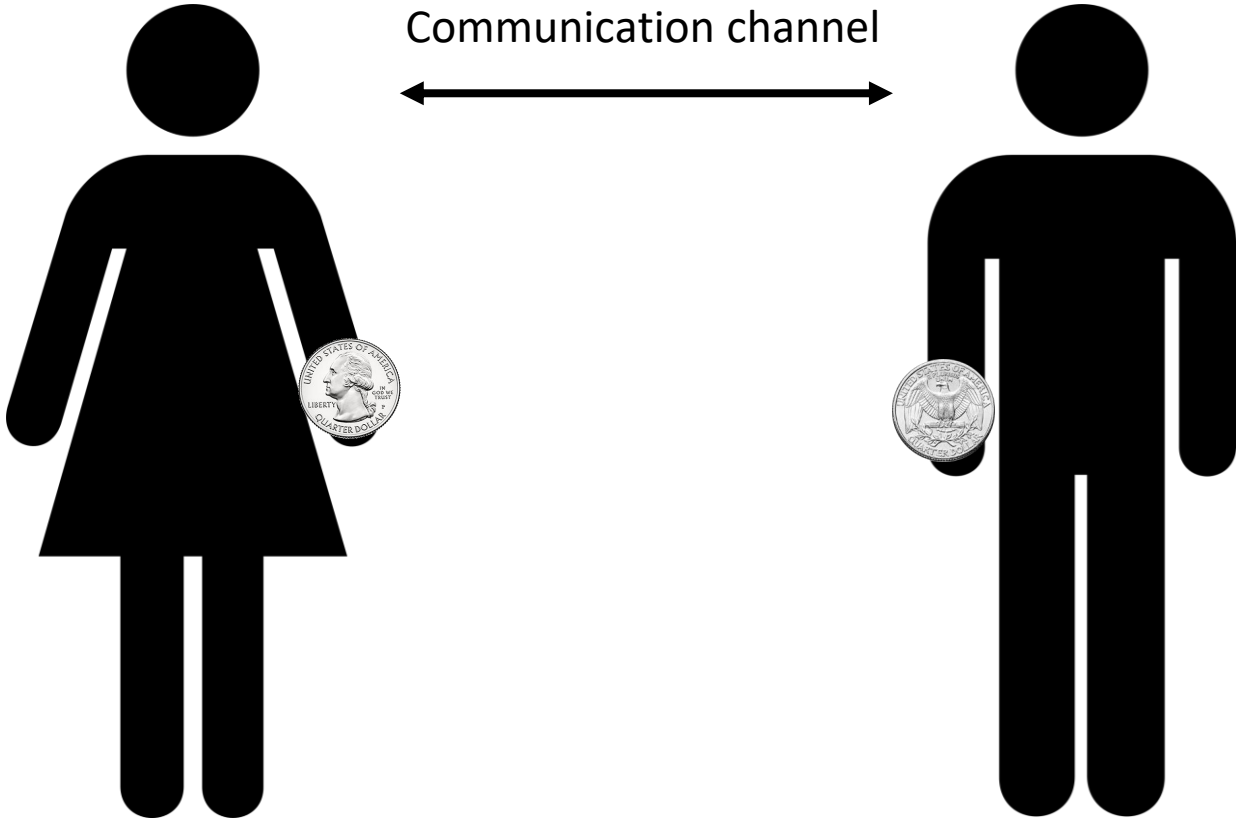
Communication complexity of equality

- $\mathrm{EQ}_n : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$

$$\mathrm{EQ}_n(x,y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

Theorem: Every deterministic communication protocol that computes EQ_n has cost at least $n + 1$

Randomized communication complexity



Randomized communication complexity of EQ_n

- Let $\delta > 0$ be any constant

Theorem: For every $n \in \mathbb{N}$, there exists a randomized communication protocol with cost $O(\log n)$ that computes EQ_n with error probability δ

- Randomized protocols are exponentially better than deterministic protocols!
- Proof: Next five slides

Randomized protocol for $\mathbf{EQ}_n$

- Think of $x, y \in \{0,1\}^n$ as numbers $x, y \in \{0, 1, \dots, 2^n - 1\}$
- Let $p_1 \leq p_2 \leq p_3 \leq \cdots$ be the sequence of all prime numbers
- **Protocol:**
  1. Alice picks $i \in \{1, 2, \dots, n/\delta\}$ uniformly at random (WLOG, n/δ is a power of two)
  2. Alice sends i and $x \bmod p_i$
  3. Bob sends a bit indicating whether $x \bmod p_i = y \bmod p_i$
  4. If so, they accept, otherwise, they reject

Analysis of the protocol: Correctness

Protocol:

1. Pick $i \in \{1, 2, \dots, n/\delta\}$ u.a.r.
2. Send i and $x \bmod p_i$
3. Check whether $x \equiv y \bmod p_i$

- If $x = y$, then $\Pr[\text{accept}] = \Pr[x \equiv y \bmod p_i] = 1$ ✔
- If $x \neq y$, then $\Pr[\text{accept}] = \Pr[x \equiv y \bmod p_i] = \Pr[p_i \text{ divides } |x - y|]$
- Let BAD be the set of prime numbers that divide $|x - y|$
- $2^{|\text{BAD}|} \leq \prod_{p \in \text{BAD}} p \leq |x - y| < 2^n$
- $\Pr[\text{accept}] = \Pr[p_i \in \text{BAD}] \leq \frac{|\text{BAD}|}{n/\delta} < \frac{n}{n/\delta} = \delta$ ✔

Analysis of the protocol: Efficiency

Protocol:

1. Pick $i \in \{1, 2, \dots, n/\delta\}$ u.a.r.
2. Send i and $x \bmod p_i$
3. Check whether $x \equiv y \bmod p_i$

- Sending i costs $O(\log n)$ bits of communication ✔
- Sending $x \bmod p_i$ costs $O(\log p_i)$ bits of communication
- How big is p_i (the i-th prime)?

Chebyshev's Estimate: Let p_k be the k-th prime. Then $p_k = O(k \cdot \log k)$.

- Therefore, $\log p_i = \log\big(O(n \cdot \log n)\big) = \log\big(o(n^2)\big) = O(\log n)$ ✔
- All that remains is to prove Chebyshev's Estimate… (Next two slides)

Step 1: Legendre's Formula

Legendre's Formula: For any $M \in \mathbb{N}$ and any prime p, the exponent of p in the prime factorization of $M!$ is precisely $\sum_{i=1}^{\infty} \lfloor M/p^i \rfloor$.

- **Proof:** Among the numbers $1, 2, 3, \ldots, M$:
  - $\lfloor M/p \rfloor$ of them are multiples of p
  - $\lfloor M/p^2 \rfloor$ of them are multiples of p^2
  - $\lfloor M/p^3 \rfloor$ of them are multiples of p^3
  - Etc.

Proof of Chebyshev's Estimate

- Let $M = \lceil 2k \cdot \log k \rceil \leq 0.5 \cdot k^2$ and let $p_1^{e_1} \cdot p_2^{e_2} \cdots p_\ell^{e_\ell} = \binom{2M}{M} = \frac{(2M)!}{(M!)^2}$

$$\text{Legendre} \Rightarrow e_j = \sum_{i=1}^{\infty} \left(\left\lfloor \frac{2M}{p_j^i} \right\rfloor - 2 \left\lfloor \frac{M}{p_j^i} \right\rfloor \right) \leq \log_{p_j}(2M), \qquad \text{so } p_j^{e_j} \leq 2M$$

- Therefore, $2^M \leq \binom{2M}{M} \leq (2M)^\ell = 2^{\ell \cdot \log(2M)}$, i.e., $\ell \geq M / \log(2M) \geq k$
- Therefore, $p_k \leq p_\ell \leq 2M = O(k \cdot \log k)$

Randomized comm

- Let $\delta > 0$ be any constant

Which of the following best describes the protocol?

A: The protocol succeeds on most pairs of inputs

B: The amount of communication is rarely more than $O(\log n)$

C: For every pair of inputs, the protocol is likely to succeed

D: It is likely that for every pair of inputs, the protocol succeeds

Respond at PollEv.com/whoza or text "whoza" to 22333

Theorem: For every $n \in \mathbb{N}$, there exists a randomized communication protocol with cost $O(\log n)$ that computes EQ_n with error probability δ

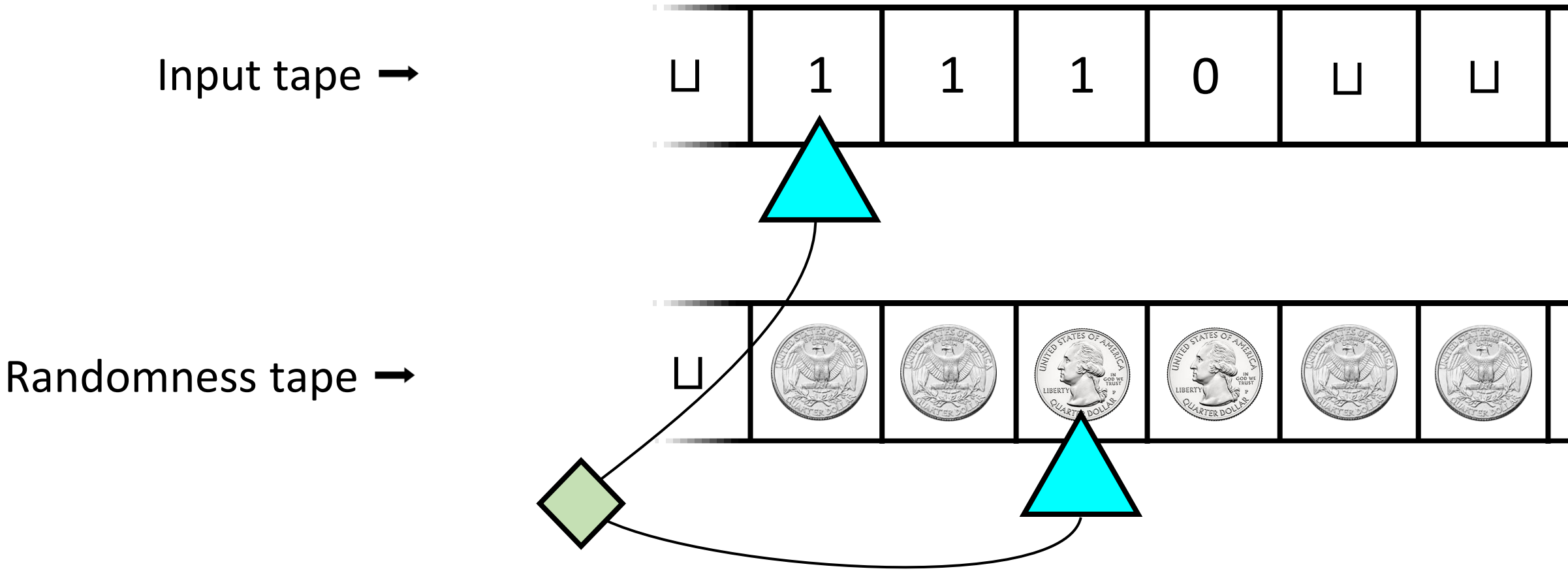
- Randomized protocols are exponentially better than deterministic protocols!

Which problems

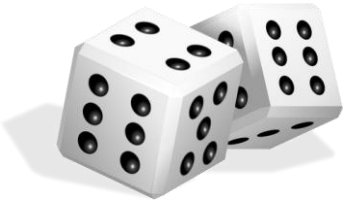
can be solved

through computation?

Randomized Turing machines

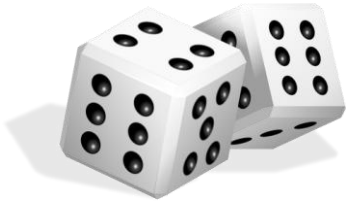


Randomized Turing machines



- Let $T: \mathbb{N} \to \mathbb{N}$ be a function (time bound)
- **Definition:** A randomized time-T Turing machine is a two-tape Turing machine M such that for every $n \in \mathbb{N}$, every $w \in \{0,1\}^n$, and every $u \in \{0,1\}^{T(n)}$, if we initialize M with w on tape 1 and u on tape 2, then it halts within $T(n)$ steps
- Interpretation: w is the input and u is the coin tosses
- (Giving M more than $T(n)$ random bits would be pointless)

Acceptance probability



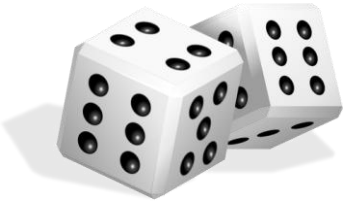
- Let M be a randomized Turing machine and let $w \in \{0,1\}^*$
- To run M on w, we select $u \in \{0,1\}^{T(n)}$ uniformly at random and initialize M with w on tape 1 and u on tape 2

$$\Pr[M \text{ accepts } w] = \frac{|\{u : M \text{ accepts } w \text{ when tape 2 is initialized with } u\}|}{2^{T(n)}}$$

Randomized TMs: Deciding a language

- Let M be a randomized time-T Turing machine for some $T: \mathbb{N} \to \mathbb{N}$
- Let $Y \subseteq \{0,1\}^*$ and let $\delta \in [0,1]$
- We say M decides Y with error probability δ if for every $w \in \{0,1\}^*$:
  - If $w \in Y$, then $\Pr[M \text{ accepts } w] \geq 1 - \delta$
  - If $w \notin Y$, then $\Pr[M \text{ accepts } w] \leq \delta$

The complexity class BPP



- **Definition:** BPP is the set of languages $Y \subseteq \{0,1\}^*$ such that there exists a randomized polynomial-time Turing machine that decides Y with error probability $1/3$

- "Bounded-error Probabilistic Polynomial-time"

Example: High school algebra

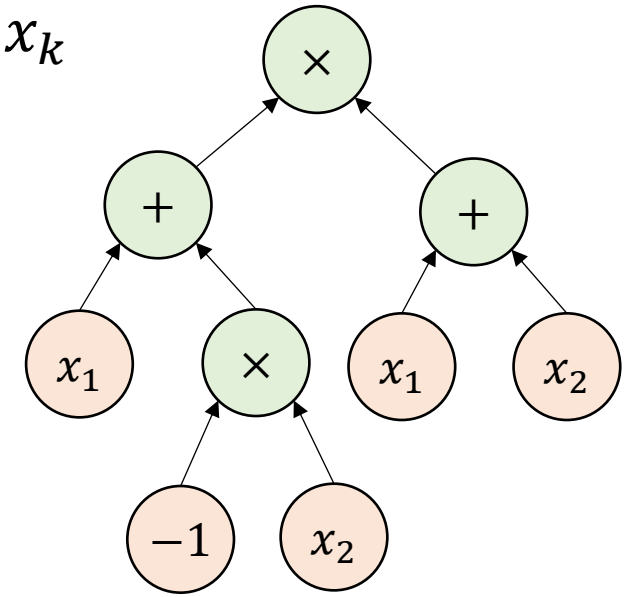
- "Expand and simplify: $\underbrace{(x+1)\cdot(x-1)}$"

This type of expression is called an arithmetic formula

- How difficult is this type of exercise?

Arithmetic formulas

- **Definition:** A k-variate arithmetic formula is a rooted binary tree
  - Each internal node is labeled with $+$ or $\times$
  - Each leaf is labeled with 0, 1, -1, or a variable among $x_1, \ldots, x_k$
- It computes $F: \mathbb{R}^k \to \mathbb{R}$
- E.g., $F(x_1, x_2) = (x_1 - x_2) \cdot (x_1 + x_2)$



Polynomial identity testing

- **Problem:** Given an arithmetic formula F, determine whether $F \equiv 0$
- **As a language:** $\mathrm{PIT} = \{\langle F \rangle : F \text{ is an arithmetic formula and } F \equiv 0\}$
- **High school algorithm:** Expand F into monomials, then simplify by canceling like terms

What is the time complexity of this algorithm?

A: $\mathrm{poly}(n)$

B: $2^{\Omega(n)}$

C: $O(1)$

D: ∞

Respond at PollEv.com/whoza or text “whoza” to 22333

Example: Polynomial identity testing

- Given: $F = (a-1)\cdot(1+b)\cdot(a-c)\cdot(1-d)\cdot(b-e)+(b+1)\cdot(d-1)\cdot(b-e)\cdot(c-a)\cdot(1-a)$
- Expand:

$$F \equiv a^2b - a^2be - a^2db + a^2dbe - acb + acbe + acdb - acdbe + a^2b^2 - a^2b^2e - a^2bdb + a^2bdbe$$
$$- abc^2 + abc^2e + abcdb - abcdbe - ab + abe + adb - adbe + cb - cbe - cdb + cdbe - b^2$$
$$+ b^2e + bdb - bdbe + b^2c - b^2ce - bcd + bcde + b^2dc - b^2dca - bdec + bdeca - b^2c + b^2ca$$
$$+ bec - beca + bdc - bdca - dec + deca - bc + bca + ec - eca - ab^2dc + ab^2dca + abdec$$
$$- abdeca + ab^2c - ab^2ca - abec + abeca - abdc + abdca + adec - adeca + abc - abca - aec$$
$$+ aeca$$

- Everything cancels out: $F \equiv 0$

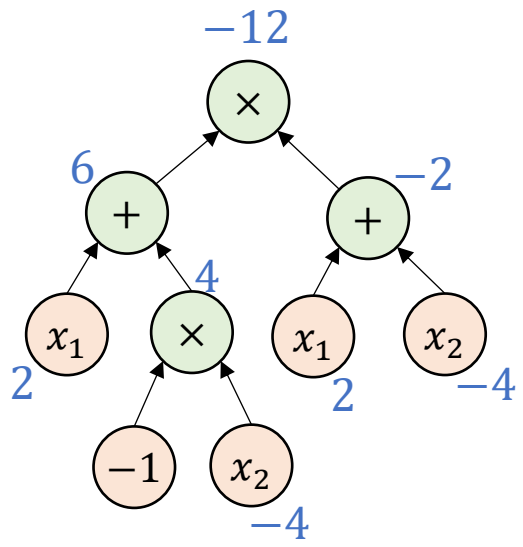
Polynomial identity testing

- Expanding F takes $2^{\Omega(n)}$ time in some cases 😟
- E.g., $F = (x + y) \cdot (x + y) \cdot (x + y) \cdots (x + y)$
- **Open Question:** Is $\mathrm{PIT} \in \mathrm{P}$?
- Next 10 slides: We will prove $\mathrm{PIT} \in \mathrm{BPP}$

Evaluating an arithmetic formula

- Let F be a k-variate arithmetic formula and let $\vec{x} \in \mathbb{Z}^k$

Lemma: Given $\langle F, \vec{x} \rangle$, one can compute $F(\vec{x}) \in \mathbb{Z}$ in polynomial time.



$x_1 = 2$

$x_2 = -4$

$F(x_1, x_2) = -12$ ✔